



UNITÉ DE RECHERCHE
INRIA-RENNES

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Volveau
Rocquencourt
BP 105
78153 Le Chesnay Cedex
France

Tél (1) 39 63 55 11

Rapports de Recherche

N° 750

**UNE APPROCHE ORIENTEE
OBJET POUR LA
REPRESENTATION ET LA
MANIPULATION DES DOSSIERS
MEDICAUX**

**Roger NOUSSI
Patrick BOSC
Sophie ROBIN**

NOVEMBRE 1987

Campus Universitaire de Beaulieu
35042 - RENNES CÉDEX
FRANCE
Téléphone : 99 36 20 00
Télex : UNIRISA 950 473 F
Télécopie : 99 38 38 32

Roger Noussi, Patrick Bosc, Sophie Robin

UNE APPROCHE ORIENTÉE OBJET POUR LA REPRÉSENTATION ET LA MANIPULATION DES DOSSIERS MÉDICAUX

Publication Interne n° 374 - Octobre 1987 - 18 pages.

RESUME

Les formalismes orientés objet sont aujourd'hui de plus en plus utilisés en intelligence artificielle. Leur succès est en partie dû aux facilités qu'ils offrent pour la spécification et l'utilisation de types abstraits de données, couvrant ainsi une partie du champ traditionnellement laissé aux bases de données. C'est aussi ce qui a motivé le choix de SHIRKA, (un système basé sur la représentation centrée objet) pour la spécification des connaissances dans GIDE [NOUSSI 86]. Dans cet article, en partant d'une rapide comparaison entre les bases de données relationnelles et les représentations centrées objet, nous montrons qu'elles offrent des outils comparables pour la description de l'aspect statique des données, mais qu'en plus les représentations centrées objet permettent à l'aide du mécanisme de filtrage et des attachements procéduraux de spécifier la dynamique du système en tenant le plus grand compte des spécificités de chaque objet. Ce comportement les rapproche dans une certaine mesure des règles de production. L'argumentation est illustrée tout au long de l'article par des exemples tirés de la maquette en cours de développement.

mots clés : représentation centrée objet, base de données, filtrage, attachements procéduraux

AN OBJECT ORIENTED APPROACH FOR THE SPECIFICATION AND THE MANIPULATION OF MEDICAL FILES

ABSTRACT

Object oriented formalisms are gaining growing success in the fields of artificial intelligence. This success is due partly to the facilities they offer in the description of abstract data types, thus covering part of the domain traditionally left to data base systems. This is the reason that motivated the choice of SHIRKA (an object centered representation system) for the implementation of GIDE system tools [NOUSSI 86]. In this paper, after a quick comparison between relational data base system and object centered representation system, we show that although they offer comparable tools for the representation of the static aspect of data, the object centered representation systems provide through procedure attachment and pattern matching mechanism more powerful tools than data base for the specification of the dynamic aspect of data.

SOMMAIRE

	page
1 Introduction	1
2 Les problèmes de la manipulation d'un dossier	2
3 Analyse des B.D. relationnelles et des R.C.O. par rapport aux besoins de GIDE	3
3.1 Description des structures des donnee dans les relation et les objets	3
3.2 Description des moyens de traitemet dans les relations et les objets	5
3.3 Le cas du système GIDE	6
4 Présentation de Shirka	7
4.1 Structure d'un schéma	7
4.2 Le filtrage	8
5 Structuration des connaissances dans G I D E	9
5.1 Les outils d'inférence	11
5.1.1 L'héritage	11
5.1.2 Les attachements procéduraux	12
5.1.3 Règles de production vues comme une combinaison du mécanisme de filtrage et des attachements procéduraux	13
5.2 Expression des fonctionnalités du système	15
5.2.1 Spécification de la fonctionnalité "saisie"	15
5.2.2 Spécification de la fonctionnalité "résumé"	16
6 Conclusion	16



UNE APPROCHE ORIENTEE OBJET POUR LA REPRESENTATION ET LA MANIPULATION DES DOSSIERS MEDICAUX

1 INTRODUCTION

Le projet G I D E (Gestion intelligente des dossiers médicaux en épilepsie) a pour but la conception d'un environnement permettant un accès rapide et intelligent au dossier d'un malade en épilepsie. La notion de rapidité utilisée ici s'apparente au fait que le système ne s'attarde pas sur certains éléments figurant pourtant dans le dossier, et l'aspect intelligent tient au fait que le système détermine lui-même si une information présente assez d'intérêt pour être fournie à l'utilisateur.

Un tel environnement doit permettre à la fois de décrire les entités du dossier dans leur aspect statique et structurel, et de réaliser les inférences nécessaires pour faire face à l'absence d'informations n'ayant pu être obtenues auprès du malade ou de son entourage. Il doit également permettre d'intégrer les éléments procéduraux servant à traiter les informations contenues dans le dossier pour en donner une présentation expressive. Par exemple, le système doit offrir plusieurs modes de présentation de la fréquence des crises (moyenne arithmétique, courbe, histogramme...), à utiliser suivant l'état du dossier.

Dans cet article, après avoir présenté la structure du dossier d'un malade qui semble correspondre aux entités représentées dans les SGBD relationnels, nous montrons comment les traitements envisagés sont plus facilement réalisables dans le cadre d'une approche orientée objet. L'argumentation sera illustrée tout au long de l'article par des exemples d'objets. Ces exemples sont tirés de la maquette en cours de développement. Cette maquette est développée à l'aide de SHIRKA [RECHE 86], un formalisme de représentation centrée objet, (R.C.O.) développé par l'INRIA à Grenoble.

2 LES PROBLEMES DE LA MANIPULATION DU DOSSIER

Les problèmes posés par la manipulation du dossier médical d'un malade en épilepsie sont liés à la précarité des informations qu'il contient et à la difficulté d'en avoir une vision exhaustive.

A) PRECARITE DES INFORMATIONS MANIPULEES

La plupart des informations contenues dans le dossier voient leur validité et même leur intérêt changer avec le temps et l'acquisition de nouvelles informations. Si on considère par exemple le cas d'un patient de moins de 6 ans que le médecin revoit après une année; même si son dossier contient des indications sur son poids (le poids du patient), on ne peut pas les considérer comme valides car ce poids a vraisemblablement beaucoup changé depuis la dernière consultation.

B) DIFFICULTE A AVOIR UNE VISION EXHAUSTIVE DU DOSSIER

Ceci est dû à la multiplicité des sources d'informations et au fait que l'acquisition de nouvelles informations n'implique pas forcément que les autres soient devenues inutiles. C'est par exemple le cas de la sémiologie dont on doit recueillir les éléments dans l'entourage du patient, et lors des consultations.

C) BUT DU SYSTEME

Habituellement, chaque consultation commence par une phase préliminaire qui peut parfois être très longue et au cours de laquelle le médecin fait la synthèse du dossier. Malgré le temps mis pour réaliser cette synthèse, on ne peut pas garantir que tous les éléments significatifs du dossier aient été pris en compte.

On peut résumer les objectifs du système G I D E de la manière suivante:

- représenter les entités composant le dossier d'un malade,
- permettre un parcours sélectif du dossier,
- permettre de réaliser sur chaque entité du dossier des traitements tenant le plus grand compte de ses particularités.

Pour accéder au dossier du malade, l'utilisateur (le médecin) n'aura pas à indiquer explicitement l'information dont il a besoin car dans la plupart des cas, il n'a pas une vision globale du dossier. Il se contente de décrire dans un jeu de question-réponse avec le système, le "contexte" dans lequel il utilise le système:

il indique par exemple si le patient est présent ou non, s'il s'agit d'une consultation de routine ou s'il souhaite remettre en cause le diagnostic ou la thérapeutique etc...

Il revient alors au système d'apprécier quelles informations du dossier doivent être communiquées à l'utilisateur.

3 ANALYSE DES B.D. RELATIONNELLES ET R.C.O. PAR RAPPORT AUX BESOINS DE GIDE

Les objectifs du système GIDE tels que résumés à la fin du paragraphe précédent semblent assez proches de ceux d'un SGBD relationnel tels qu'énumérés dans [GARDA 86] et résumés ci-dessous:

- décrire les structures de données rangées dans la base,
- définir des propriétés additionnelles ou contraintes d'intégrité,
- définir un ensemble d'opérations permettant de manipuler les données.

Bien que ce rapprochement soit réel sur les deux premiers points, la différence est fondamentale sur le troisième et de ce fait les solutions exigées s'apparentent mieux aux techniques offertes par les R.C.O., comme nous allons le montrer.

3.1 DESCRIPTION DES STRUCTURES DE DONNEES DANS LES RELATIONS ET LES OBJETS

Sur le plan de la description des structures de données, on peut faire le rapprochement entre le concept de relation et le concept d'objet. En effet, de même qu'une relation est caractérisée par son nom, la liste de ses constituants et leur domaines, un objet est caractérisé par son nom, la liste de ses attributs, les attributs étant eux-mêmes décrits par des facettes dont certaines servent justement à préciser le domaine des valeurs des attributs. Jusqu'à ce point, la différence semble être due à la terminologie; Si on considère par exemple le concept d'épilepsie, il peut être décrit par la relation et l'objet ci-dessous:

a) relation EPILEPSIE

EPILEPSIE(TE, AGE-DEBUT, FF, FAC-F, ATCD, CRISES)

TE=type d'épilepsie

FF=foyers focaux

FAC-F=facteurs

ATCD=antécédents

Dans le dictionnaire des données, on précisera le domaine des valeurs des attributs et les contraintes d'intégrité attachées à cette relation.

b) l'objet EPILEPSIE (suivant une syntaxe proche de celle de SHIRKA)

```
(def-sh '(EPILEPSIE
          (TE                ($un chaine)
                               ...    ))
          (AGE-DEBUT         ($un entier)
                               ($intervalle(0 120))
          (FF                ... )
          (FAC-F             ... )
          (ATCD              ... )
          (CRISES            ... ))))
```

FIGURE 1

Pour l'instant, nous insistons sur les attributs en faisant abstraction des facettes qui complètent cet objet.

A ce niveau, l'utilisation de la relation EPILEPSIE pose un certain nombre de problèmes:

- elle n'est pas dans la première forme normale; en effet, les composantes FF et FAC-F peuvent prendre simultanément plusieurs valeurs.

Son stockage sous cette forme poserait des problèmes de redondance et de mise à jour dans la base de donnée.

- la composante CRISES n'est pas une composante élémentaire; le concept de crise peut être lui-même défini comme une relation avec plusieurs constituants.

Pour être stockée, cette relation doit subir un certain nombre de transformations.

Les problèmes évoqués ci-dessus sont inhérents à la préoccupation de stockage optimum des SGBD et à la nécessité d'éviter les redondances qui peuvent induire des incohérences dans la base, notamment après les mises à jour. Dans les R.C.O. par contre la préoccupation est de regrouper dans une même unité syntaxique les éléments qui se rattachent à un concept donné du domaine d'application. On cherche à saisir la signification intrinsèque de l'objet en dehors de tout problème de redondance ou de stockage. Cette différence se répercute au niveau du choix des relations à faire figurer dans la base ou des objets à définir dans le système. En base de données, il existe des algorithmes pour obtenir une décomposition sans perte à partir d'une table représentant une relation logique.

Dans les formalismes orientés objet on se fie à l'expérience d'un expert qui a une bonne connaissance des concepts à définir. Sans résoudre le problème de redondance et d'optimisation d'espace, ceci permet d'appréhender un certains nombre d'éléments plus liés à la sémantique de l'objet qu'à sa structure et qui sont d'une importance déterminante pour l'expression de la dynamique du système; il s'agit notamment de:

- l'organisation des objets en classes et sous-classes avec utilisation du concept

d'héritage,

- la détermination des valeurs par défaut,
- la définition des procédures à attacher à un objet.

La différence entre les bases de données relationnelles et les R.C.O. est essentiellement basée sur ces trois éléments.

3.2 DESCRIPTION DES MOYENS DE TRAITEMENT DANS LES OBJETS ET LES RELATIONS

Pour les R.C.O., il y a deux aspects à distinguer dans le cadre d'une comparaison avec les bases de données:

- l'aspect descriptif lié aux concepts d'objet, de classe et d'héritage,
- l'aspect dynamique lié aux attachements procéduraux et au mécanisme de filtrage.

La similitude avec les bases de données ne peut se faire que sur le premier point car contrairement aux formalismes objet où il y a une forte intégration entre les aspects descriptif et dynamique, les différents modèles de base de données ne mettent l'accent que sur l'aspect descriptif, la manipulation des données étant faite par un langage de manipulation bien distinct du langage de description.

Dans les R.C.O. l'expression de la sémantique et de la dynamique de l'objet se fait à travers les moyens de traitement qui lui sont spécifiques et qu'on lui rattache à l'aide des facettes. Les procédures de traitement sont réparties dans les objets auxquels elles s'appliquent. La seule opération appliquée globalement est le mécanisme de filtrage qui à partir de la forme syntaxique d'un objet recherche à travers le treillis des classes celle à laquelle appartient l'objet, permettant ainsi de déterminer les moyens de traitement qui lui sont adaptés.

la représentation du schéma DOSSIER à l'aide de SHIRKA peut se résumer de la manière suivante:

```
(def-sh '(DOSSIER
          (sorte-de      (= objet ))
          (NUMERO-DOSSIER ($un entier))
          (CONTEXTE      ( ... ))
          (PATIENT        ($un personne)
                        ($sib-exec inst-personne)
                        ($si-succes ranger-valeur)
                        ($si-echec imp-lib1)
                        ($pour-vi
                          (imp-homme)
                          (imp-femme)
                          (imp-enfant)))
          (EPILEPSIE      ... )))
```

fig. 2

A l'attribut PATIENT sont rattachés un certain nombre de moyens de traitement qui sont propres à un schéma de type PERSONNE. Les procédures `inst-personne`, `ranger-valeur` et `imp-lib1` introduites par les facettes `$sib-exec`, `$si-succes` et `$si-echec` permettent respectivement de:

- évaluer l'attribut PATIENT quand il n'est pas connu,
- sauvegarder la valeur obtenue lorsqu'elle a été caculée avec succès,
- fournir une alternative si le calcul n'a pu se faire.

La facette `$pour-vi` permet d'introduire les procédures utilisables pour imprimer l'attribut PATIENT. Dans ce cas précis le processus de filtrage permettra d'abord de déterminer la sous-classe de la classe PERSONNE à laquelle appartient le schéma à imprimer, et selon que ce schéma sera de la sous-classe HOMME, FEMME, ENFANT, choisira le traitement approprié.

Dans les SGBD relationnels toutes les opérations prévues (opérations d'algèbre relationnelles) ne sont pas propres à une relation ou une catégorie de relation, mais s'appliquent de manière indifférente à toutes les relations. La dynamique associée aux relations est complètement déconnectée de la description de celles-ci. Ceci oblige à mettre plutôt dans les programmes d'utilisation des éléments qui sont naturels au modèle mais qui ne peuvent s'appliquer indifféremment à toutes les relations. Par exemple, les procédures d'instanciation et d'impression évoquées dans le schéma ci-dessus ne peuvent être intégrées dans un modèle relationnel.

Les spécificités des différentes relations ne sont pas prises en compte.

3.3 LE CAS DU SYSTEME GIDE

Comme nous l'avons déjà dit, le médecin n'a pas toujours une bonne vision globale du dossier. Le système doit alors saisir toutes les nuances des informations à manipuler pour pouvoir les utiliser dans une requête qui est en principe assez vague puisque le médecin n'indique pas explicitement l'information dont il a besoin. La principale raison qui fait penser qu'une base de données ne serait pas adaptée au problème posé dans GIDE est que la forme des requêtes impose que la description des données intègre beaucoup d'informations sémantiques où relevant de l'aspect dynamique pour que le système puisse y répondre. En outre, la quasi-totalité des traitements envisagés dans GIDE impliquent un seul dossier et non l'ensemble des dossiers du service. Le problème d'espace utilisé pour la représentation n'est pas à priori contraignant. Le fait que la masse d'information à gérer dans une séance de travail ne soit pas vraiment énorme contribue à faire penser qu'une approche orientée objet s'adapte mieux à ce genre de problème.

Comme on peut le constater, la description des connaissances dans GIDE exige une forte intégration entre l'aspect descriptif et l'aspect dynamique des connaissances impliquées. Cet objectif peut en réalité être atteint par tous les

formalismes basés sur la notion d'objet (langages orientés objets et représentations centrées objets). En outre, la collaboration avec les experts en épilepsie a montré qu'il était important de pouvoir utiliser des connaissances en petits modules organisées dans le style des règles de production. Le fait que les R.C.O. en général et SHIRKA (que nous présentons au paragraphe suivant) en particulier offre un mécanisme de filtrage qui combiné avec les attachements procéduraux permette de reproduire l'effet des règles de production (cf paragraphe 5.3.3) nous a poussé à choisir celui-ci pour l'implémentation de la maquette.

4 PRESENTATION DE SHIRKA

Shirka est un système de gestion de base de connaissances centrées-objet qui utilise un modèle de représentation inspiré des FRAMES [MINSK 75]. Une entité dans SHIRKA est appelée schéma. Un schéma représente une classe d'objets ou un objet particulier. Le terme objet ici désigne une entité physique ou conceptuelle, une situation, un contexte, ou toute autre entité sémantiquement bien définie. Il y a aussi les schémas "méthode" qui permettent d'exprimer les attachements procéduraux et dont l'un des attributs est la procédure à appeler en cas de succès d'instantiation du schéma.

4.1 STRUCTURE D'UN SCHEMA

Un schéma est décrit par son nom et la liste de ses attributs. Chaque attribut est à son tour décrit par un nom et une liste de facettes. Les facettes sont choisies dans un ensemble prédéfini qui ne peut être étendu par l'utilisateur.

Si on se propose par exemple de décrire le schéma DATE, on peut le caractériser par les attributs JOUR, MOIS, ANNEE. Chaque attribut sera à son tour décrit par des facettes qui sont des primitives de SHIRKA. On aura donc pour DATE la description ci-dessous:

```
(def-sh '(DATE
          (sorte-de      (= objet))
          (JOUR          ($un entier)
                    ($intervalle(1 30))
          (MOIS          ($un chaine)
                    ($domaine "janvier" "février. . ." "décembre"))
          (ANNEE         ($un entier))))
```

figure 3 définition du schéma DATE

Dans cette description, les facettes utilisées sont *\$un*, *\$intervalle* *\$domaine*.

\$un permet d'introduire le type de l'attribut en précisant qu'il est mono-valué; un attribut multi-valué aurait été introduit par *\$liste-de*.

\$intervalle permet de préciser l'intervalle dans lequel l'attribut prend ses valeurs.

\$domaine permet d'introduire la liste des valeurs possibles de l'attribut.

Shirka possède plus d'une trentaine de facettes qui peuvent être réparties dans les différents groupes suivants:

- les facettes de type ou de restriction de type(*\$intervalle*,*\$domaine* en font partie),
- les facettes de détermination des valeurs d'attribut,
- les facettes de réaction sur événement,
- les facettes de visualisation,
- les facettes de lecture,

4.2 LE FILTRAGE

Le filtrage permet de déterminer si une instance appartient à une classe, afin de savoir si on peut la donner comme valeur à un attribut de type donné, ou si on peut lui appliquer certains traitements. Il utilise deux paramètres dont l'un est l'instance à classer et l'autre la classe(le filtre) par rapport à laquelle il faut situer l'instance. Le filtre est décrit par un schéma de classe qui porte le nom de la classe à laquelle se rattachent les instances cherchées. Sa structure en est identique mais chacun de ses attributs peut posséder des restrictions supplémentaires par rapport à celles figurant dans le schéma de classe.

Exemple

(DATE

(jour (\$valeur 25)))

Cet exemple décrit un filtre qui permet de retrouver l'ensemble des instances du schéma DATE dont la composante JOUR a pour valeur 25.

Dans la fig. 9 (p. 12), la facette *\$sib-filtre* introduit un filtre qui sera utilisé pour retrouver les instances de l'objet médecin dont l'attribut **intervention** a pour valeur '*traitant*' et la donnera comme valeur à l'attribut **med-traitant** du schéma ENV-HOSP.

5 STRUCTURATION DES CONNAISSANCES DANS GIDE

L'entité la plus complexe du système c'est l'entité consultation schématisée ci-dessous:

CONSULTATION

- contexte
- dossier

les attributs du schéma CONSULTATION sont eux-mêmes des schémas complexes. Une séance de travail avec GIDE consiste à instancier le schéma CONSULTATION ou à compléter une instance dont l'évaluation a commencé à la précédente consultation.

La composante contexte doit être complètement réévaluée à chaque consultation; les valeurs de ses attributs ne sont pas sauvegardées d'une consultation à l'autre. Quant aux objets dont l'instantiation de l'attribut DOSSIER implique l'utilisation, leur évaluation au cours d'une consultation dépend de plusieurs éléments; entre autre:

- s'ils avaient été évalués lors de la précédente consultation,
- si le système prévoit la sauvegarde des valeurs obtenues par leur instantiation,
- si l'instantiation d'autres schémas ne met pas en défaut des contraintes d'intégrité qui les impliquent; auquel cas, les anciennes valeurs sont détruites et réévaluées en cas de besoin.

L'attribut DOSSIER du schéma consultation est lui-même un schéma complexe décrit à son tour par les attributs ci-dessous

DOSSIER

- patient
- prise en charge
- épilepsie
- suivie

Pour préciser l'utilisation des schémas définis dans le système, nous allons détailler l'attribut PATIENT (un patient est considéré comme une sorte de personne) du schéma DOSSIER. Le schéma PERSONNE est défini à partir du schéma IDENTITE ci-dessous:

```
(def-sh '(IDENTITE
      (sorte-de      (= objet))
      (NOM-PRENOM    ($un chaine))
      (ADRESSE-PATIENT ($un adresse)
                ($sib-exec inst-adresse)
                ($sib-succes ranger-valeur))
      (TELEPHONE     ($un chaine))))
```

Figure 4 définition du schéma IDENTITE

Cette définition comporte deux aspects:

- un aspect statique et descriptif portant sur la structure de l'objet et qui est traduit par la donnée des attributs *nom-prénom*, *adresse-patient* et *téléphone*. Désormais le système reconnaît dans le type structuré IDENTITE, une agglomération des types plus simples *nom-prénom*, *adresse-patient* et *téléphone*.
- Un aspect dynamique permettant au système de manipuler la structure une fois celle-ci constituée et qui dans le cas de la composante adresse-patient est introduit par les facettes \$sib-exec et \$si-succès. Ces facettes permettent respectivement de décrire les traitements à exécuter si la valeur de l'objet est inconnue et si son évaluation est entreprise avec succès.

```
(def-sh '(ADRESSE
      (sorte-de      (= objet))
      (NUMERO        ($un entier))
      (RUE           ($un chaine))
      (CODE-POSTAL   ($un entier))
      (VILLE        ($un chaine))))
```

Figure 5 description du schéma adresse.

```
(def-sh '(DATE
      (sorte-de      (= objet))
      (lui-meme      ($a-verifier concordance-jour-mois))
      (JOUR          ($un entier)
                ($intervalle(1 31))
      (MOIS          ($un chaine)
                ($domaine "janvier" . . ."decembre")
                ($pour-vi imp-chaine))
      (ANNEE         ($un entier))))
```

Figure 6 Dans le schéma DATE, la contrainte concordance-jour-mois qui implique plusieurs attributs est rattachée directement à l'entité DATE désignée par l'attribut "lui-meme" qui est implicite dans tous les schémas.

```

(def-sh '(PERSONNE
        (sorte-de      (= identite))
        (lui-meme      ($pour-vi (imp-homme)
                                   (imp-femme)
                                   (imp-enfant)))
        (SEXE          ($un chaine)
                        ($domaine "masculin" "féminin"))
        (PROFESSION    ($un chaine))
        (AGE            ($un entier)
                        ($sib-exec
                          (compter-annee
                           (date-origine<- date-naiss)
                           (nb-annee-> age))))

        (DATE-NAISS    ($un date)
                        ($sib-exec inst-date)
                        ($si-succes ranger-valeur))))

```

Figure 7 description du schéma PERSONNE

5.1 LES OUTILS D'INFERENCE

Shirka offre un certain nombre d'outils d'inférence dont il est fait usage dans GIDE; il s'agit notamment de l'héritage, du filtrage, des valeurs par défaut, de l'attachement procédural; d'autre part on peut reproduire le mécanisme de chaînage arrière par combinaison de plusieurs de ces outils.

5.1.1 L'HERITAGE

L'héritage est un mécanisme commun à tous les formalismes de représentation de connaissance à base d'objets. Il peut porter sur des attributs ou sur des valeurs d'attributs.

Ainsi, en plus des attributs qui lui sont propres et qui sont *sexe*, *profession*, *age* et *date-naiss*, le schéma PERSONNE (fig. 7) hérite du schéma IDENTITE (fig. 4) les attributs *nom-prénom*, *adresse* et *téléphone*.

D'autre part, dans l'instanciation du schéma GARCON (fig 8) par exemple, les valeurs des attributs *profession* et *sexe* ne seront plus demandées. Ils héritent respectivement de la valeur par défaut et de la valeur fixée par définition dans les schémas GARCON et ENFANT.

La différence entre la valeur par défaut et la valeur fixée par définition est que la valeur par défaut peut être modifiée. Ainsi on peut trouver un garçon ayant une

profession mais son sexe sera toujours "masculin".

```
(def-sh '(ENFANT
          (sorte-de           (=  personne))

          (AGE                 ($intervalle(1 15))

          (PROFESSION          ($valeur "sans  profession"))))
```

```
(def-sh '(GARCON
          (sorte-de           (=  enfant))
          (SEXE                ($valeur "masculin"))
```

fig 8 définition des concepts ENFANT et GARCON

```
(def-sh '(ENV-HOSP
          (sorte-de           (=  objet))

          (NBRE-DE-CONSULTATION ($un entier))

          (MED-TRAITANT         ($un medecin)
          ($sib-filtre
                                (medecin
                                (intervention
                                ($valeur "traitant"))))

          (MED-CONSULTANT       ($un medecin))

          (DATE-DERNIERE-CONS   ($un date))))
```

fig 9 definition du schéma ENV-HOSPITALIER

5.1.2 LES ATTACHEMENTS PROCEDURAUX

Les attachements procéduraux peuvent avoir des effets très variés suivant les facettes qui les introduisent; il peuvent servir à:

- déterminer la valeur d'un attribut (\$sib-exec, \$sib-filtre, \$sib-choix);
- attacher une contrainte à un attribut (\$à-vérifier);
- visualiser l'attribut (\$avant-vi, \$pour-vi);
- lire l'attribut.

On peut combiner les possibilités d'attachements procéduraux et de filtrage pour avoir un comportement plus flexible du système: à la fig 7 par exemple, le système choisira la méthode de visualisation adaptée selon que l'instance de PERSONNE à imprimer appartient à la sous-classe HOMME, FEMME, ou ENFANT.

Rappelons que le mécanisme de filtrage (parag. 4.2) est l'un des principaux outils d'inférence utilisés dans Shirka.

5.1.3 REGLES DE PRODUCTION VUES COMME UNE COMBINAISON DU MECANISME DE FILTRAGE ET DES ATTACHEMENTS PROCEDURAUX

Dans ce paragraphe, nous allons montrer comment on peut reproduire le mécanisme des règles de production en adjoignant plusieurs schémas méthodes à une facette; considérons la règle suivante pouvant servir à décrire le résumé de l'identité d'un patient:

SI l'âge du patient est compris entre 16 et 120 ans et
le sexe du patient est "masculin"

ALORS résumer son identité par PROC1.

Dans MYCIN [SHORT 76] par exemple on aurait dit que cette règle se rattache au contexte patient. En effet les prémisses de cette règle portent toutes sur les caractéristiques de l'objet PATIENT. Comme l'objet PATIENT existe dans le système, on peut créer une sous-classe de cette classe par spécialisation des attributs correspondant aux caractéristiques évoquées dans la règle. Cette sous-classe peut être décrite par le schéma HOMME ci-dessous:

```
(def-sh '(HOMME
  (sorte-de (= personne))
  (AGE ($domaine(16 120))
  (SEXE ($à-verifier
    (égalité
      (valeur1<- sexe))
      (valeur2<- "masculin")))))
```

fig. 10 Le schéma HOMME est un schéma qui a toutes les caractéristiques du schéma PERSONNE, mais avec des restrictions supplémentaires.

La règle ci-dessus peut être traduite par le schéma suivant

```
(def-sh '(RESUME-HOMME
  (sorte-de (= méthode))
  (malade ($un HOMME))
  (nom-fct ($valeur PROC1)))
```

fig. 11

La fonction PROC1 ne sera activée que si l'objet MALADE qui lui est transmis peut être vu comme appartenant à la sous-classe HOMME et donc vérifiant les contraintes supplémentaires portées sur HOMME par rapport à PERSONNE.

De la même façon on peut définir des règles de résumé dont les prémisses correspondent à des spécialisations de l'objet PERSONNE équivalentes à FEMME, GARCON, FILLE et dont les noms seraient respectivement RESUME-FEMME, RESUME-GARCON, RESUME-FILLE.

```
(def-sh '(RESUME-FEMME
          (sorte-de (= méthode))
          (malade ($un FEMME))
          (nom-fct ($valeur PROC2))))
```

fig. 12

Le schéma PERSONNE (fig. 7) peut être complété de la manière suivante:

```
(def-sh '(PERSONNE
          (sorte-de (= identité))
          (lui-même
            ($pour-vi
              (RESUME-HOMME(malade<- lui-même))
              (RESUME-FEMME(malade<- lui-même))
              (RESUME-GARCON(malade<- lui-même))
              (RESUME-FILLE(malade<- lui-même))))
```

-->le reste sans changement<--

fig. 13

Pour visualiser le schéma PERSONNE (facette \$pour-vi), le système associera à la composante malade des schémas de traitement mentionnés ci-dessus l'instance de PERSONNE à visualiser (référéncé ici par "lui-même"). Par le mécanisme de filtrage et classification, le système détermine à quelle sous-classe du schéma PERSONNE appartient la composante malade des schémas de résumé proposés. Selon que cet objet pourra être vu comme appartenant à la classe HOMME, FEMME, ..., les traitements PROC1.....PROC4 seront appliqués. L'utilisation de ces schémas est équivalente à celle de l'ensemble des règles ci-dessous:

- 1) SI l'âge du patient est compris entre 16 et 120 ans et
le sexe du patient est "masculin"
ALORS résumer son identité par PROC1.

- 2) SI l'âge du patient est compris entre 16 et 120 ans et
le sexe du patient est "féminin"
ALORS résumer son identité par PROC2.
- 3) SI l'âge du patient est compris entre 0 et 15 ans et
le sexe du patient est "masculin"
ALORS résumer son identité par PROC3.
- 4) SI l'âge du patient est compris entre 0 et 15 ans et
le sexe du patient est "féminin"
ALORS résumer son identité par PROC4.

5.2 EXPRESSION DES FONCTIONNALITES DU SYSTEME

Dans [NOUSS 87] la structure d'un objet susceptible de servir à la description des entités du dossier d'un malade est caractérisé par les éléments suivants:

- a) description de la morphologie de l'objet (sa classe, sa spécialisation, ses attributs),
- b) description des éléments servant à l'instanciation de l'objet.
- c) description des méthodes servant à spécifier la façon dont l'objet répond aux sollicitations de son environnement.
- d) description des qualifications servant à manipuler l'objet sans faire référence aux granules d'informations sous-jacentes.

Les éléments correspondant au point a) sont implicites dans les primitives de SHIRKA. Les éléments correspondant aux points b) et c) sont ceux qui dans l'état actuel du système permettent de remplir les fonctionnalités "saisie" (ou "évaluation") et "résumé". Ils correspondent à la description des schémas qui seront donnés comme valeur aux facettes \$sib-XXX et \$pour-vi des entités que l'on veut résumer ou saisir.

5.2.1 SPECIFICATION DE LA FONCTIONALITE "SAISIE"

L'évaluation ou la saisie d'un objet est décrite par la donnée des schémas servant de valeur pour les facettes \$sib-exec, \$sib-filtre, et \$sib-choix. Ces schémas peuvent être des méthodes qui à partir du dialogue avec l'utilisateur, créent une instance de l'objet en cours d'évaluation. Pour l'évaluation du schéma PERSONNE, on utilise le schéma de traitement suivant:

```
(def-sh 'INST-PERSONNE
  (sorte-de (= méthode))
  (résultat ($un PERSONNE))
  (nom-fct ($valeur SAISIE-PERSONNE))))
```

fig. 14 La fonction SAISIE-PERSONNE permet d'obtenir les valeurs des attributs du schéma PERSONNE qui seront ensuite affectées à RESULTAT

Les schémas utilisés comme valeurs des facettes peuvent également calculer les attributs du schéma à instancier à partir d'autres objets connus par un mécanisme semblable à celui utilisé pour calculer l'âge de PERSONNE et décrit par le schéma ci-dessous:

```
(def-sh '(INST-AGE
  (sorte-de (= méthode))
  (DATE-ORIGINE ($un date))
  (DATE-AUJORDHUI ($un date))
  (AGE ($un entier))
  (nom-fct ($valeur compter-année))))
```

fig. 15 A partir des objets *date-origine* et *date-aujourd'hui* ce schéma permet de calculer l'âge du patient en utilisant la fonction compter-année.

5.2.2 SPECIFICATION DE LA FONCTIONNALITE "RESUME"

Le résumé est considéré comme une visualisation pour laquelle les méthodes utilisées (et de ce fait le résultat) varient suivant les spécialisations des concepts impliqués et du contexte. Le cas décrit au paragraphe 5.1.3 constitue une forme de résumé car la procédure utilisée varie selon que l'instance impliquée peut être vue comme appartenant aux sous-classes HOMME, FEMME, GARCON ou FILLE. Le fait de connaître la sous-classe à laquelle appartient l'instance à résumer permet de mieux cibler les informations à fournir en guise de résumé.

6 CONCLUSION

Nous avons essayé de montrer comment une approche centrée objet était mieux adaptée que les bases de données au problème posé dans GIDE; en effet, bien qu'offrant sur le plan de la description de la structure des données des outils comparables sur beaucoup de points à ceux d'un SGBD relationnel, les R.C.O. ont l'avantage d'intégrer à la description des objets, des éléments permettant de spécifier la dynamique et de préciser la sémantique des structures décrites. De plus, les R.C.O. offrent un mécanisme de filtrage qui combiné avec les attachements procéduraux, permet de reproduire le mécanisme des règles de production, couvrant ainsi certains aspects du traitement relevant de l'intelligence artificielle. On peut cependant se

problème de la lourdeur en exécution et de l'espace mémoire nécessaires pour mettre en oeuvre un tel système. Dans le cas de GIDE, ce problème est atténué dans la mesure où le système ne manipule dans une séance de travail qu'un seul dossier et non l'ensemble des dossiers du service; le problème d'espace utilisé n'est donc pas pas à priori très contraignant. On peut imaginer une base de donnée relationnelle pour stocker l'ensemble des dossiers du service, le système GIDE servant d'interface pour prendre en main le dossier précis dont on veut faire la synthèse.

BIBLIOGRAPHIE

- [FARRE 85] H. Farreny
Les systèmes experts: principes et exemples
collection techniques avancées de l'informatique Cépadues edition.
- [GARDA 87] G. Gardarin, F. Viallet
Bases de donnée relationnelles Supra et cincom
édition Eyrolles 1987.
- [MINSK 75] M. Minsky
A framework for representing knowledge,
MIT press 1975.
- [NOUSS 87] R. Noussi, M. Courant, M., S. Robin
Formalisme de spécification des connaissances à base d'objet pour la synthèse des dossiers médicaux
IRISA, publication interne No 344 janvier 1987.
- [RECHE 85] F. Rechenman
SHIRKA: mécanismes d'inférence sur une base de connaissances centrée objet
5e congrès AFCET sur la reconnaissance des formes et l'intelligence artificielle,
Grenoble nov. 1985.
- [RECHE 86] F. Rechenman
Manuel d'utilisation de Shirka
(Document provisoire; non publié).
- [SHORT 76] E. H. Shortliffe
Computer-based medical consultation: MYCIN
American Elsevier, 1976.

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique

